IN THE CLAIMS

Claims 1, 3, 5-10, 12-17, 19-22, and 25-30 are pending in this application.

Claims 2, 4, 11, 18, 23, and 24 have been canceled.

Claims 1, 3, 5-7, 10, 17, 19-22, and 25-28 have been amended.

1. (Currently amended) A method for responding to a spurious timeout, comprising:

adjusting congestion state values by restoring a slow-start threshold value, setting a pipe value based on adding a maximum segment size capable of being sent by a sending host to a difference between a maximum sequence number sent so far and a lowest sequence number that is not yet acknowledged, and re-setting an initial value of a congestion window;

maintaining a data flow on a network in accordance with the adjusted congestion state values, wherein data is transmitted if the pipe value is less than a congestion state window; and

re-transmitting previously transmitted data when the previously transmitted data has been deemed to be lost on the network.

2. (Canceled)

3. (Currently amended) A method according to ~~Claim 2~~ Claim 1, wherein the slow-start threshold is a value of usable bandwidth detected prior to the timeout.

4. (Canceled)

5. (Currently amended) A method according to ~~Claim 2~~ Claim 1, wherein the re-setting the initial value of the congestion window includes setting the congestion window to be double that of a maximum data segment size.

6. (Currently amended) A method according to ~~Claim 2~~ Claim 1, wherein maintaining a data flow in accordance with the adjusted congestion state values includes:

transmitting a data packet;

receiving an acknowledgement; and

re-setting the congestion window by adding the maximum data segment size capable of being sent by the sending host.

7. (Currently amended) A method according to ~~Claim 2~~ Claim 1, wherein re-transmitting previously transmitted data when the previously transmitted data has been deemed to be lost on the network includes re-transmitting previously

transmitted data when three duplicate acknowledgements are received by a sending host.

8. (Original) A method according to Claim 7, further comprising implementing a slow-start recovery process.

9. (Original) A method according to Claim 7, further comprising implementing a slow-start recovery process, which includes:

readjusting the pipe value; and

re-setting the size of the congestion window in accordance with a pattern of received acknowledgements.

10. (Currently amended) A method for responding to a spurious timeout on a network, comprising:

restoring congestion state values, including setting a limit of data that a sending host can send over the network before receiving an acknowledgement;

restoring a threshold value of available bandwidth prior to the spurious timeout;

adjusting a pipe value, wherein the pipe value is based on an estimate of data outstanding on the network prior to the spurious timeout and is based on adding a maximum segment size capable of being sent by a sending host to a

difference between a maximum sequence number sent and a lowest sequence number that is not yet acknowledged;

maintaining a data flow from the sending host, wherein data is transmitted if the pipe value is less than a congestion state window; and

resetting, upon receiving an acknowledgement, the limit of data that the sending host can send over the network before receiving an acknowledgement.

11. (Canceled)

12. (Original) A method according to Claim 10, wherein the limit of data that the sending host can send over the network in before receiving an acknowledgement is set to twice the maximum data segment size that the sending host can send.

13. (Original) A method according to Claim 10, wherein the limit of data that the sending host can send over the network before receiving an acknowledgement is re-set, upon receiving an acknowledgement, by adding the maximum data segment size that the sending host can send.

14. (Original) A method according to Claim 10, further comprising re-transmitting data when data previously transmitted over the network is confirmed to be lost on the network.

15. (Original) A method according to Claim 14, wherein data previously transmitted over the network is confirmed to be lost on the network upon receiving three duplicate acknowledgements.

16. (Original) A method according to Claim 9, further comprising maintaining a data flow according to a slow-start recovery process.

17. (Currently amended) A computer-readable storage medium embedded with a computer executable program having at least one instruction that, upon detecting a timeout on a network, causes at least one processor to:

adjust congestion state values including adjusting a pipe value, wherein the pipe value is based on an estimate of an amount of data outstanding in the network to a maximum segment size sent by a sending host plus a difference between a maximum sequence number sent and a lowest sequence number that is not yet acknowledged;

maintain a data flow on the network, wherein data is transmitted if the pipe value is less than a congestion state window; and

re-transmit previously transmitted data when the previously transmitted data is determined to be lost on the network.

18. (Canceled)

19. (Currently amended) A computer-readable storage medium embedded with a computer executable program according to Claim 17, wherein the at least one instruction to adjust congestion state values includes at least one instruction to:

limit an amount of data that a sending host can send before receiving an acknowledgement to be twice a maximum data segment size capable of being sent by the sending host.

20. (Currently amended) A computer-readable storage medium embedded with a computer executable program according to Claim 19, wherein the at least one instruction to maintain a data flow on the network includes at least one instruction to:

increase the amount of data that the sending host can send before receiving an acknowledgement by the maximum data segment size capable of being sent by the sending host.

21. (Currently amended) A computer-readable storage medium embedded with a computer executable program according to Claim 17, wherein the at least one instruction to re-transmit previously transmitted data when the previously transmitted data is determined to be lost on the network includes at least one instruction that causes at least one processor to:

initiate slow-start processing.

22. (Currently amended) An apparatus for spurious timeout recovery, comprising:

a transmitter to transmit data packets;

a transmission timer to detect a spurious timeout; and

a response processor

to adjust congestion state values by setting a limit for an amount of data that a sending host sends before receiving an acknowledgement to be twice a size of a data segment that the sending host can send,

to maintain a data flow on a network with the adjusted congestion state values, wherein data is transmitted if a pipe value is less than a congestion state window, wherein the pipe value is based on adding a maximum segment size capable of being sent by the sending host to a difference between a maximum sequence number sent and a lowest sequence number that is not yet acknowledged, and

to re-transmit previously transmitted data when the previously transmitted data has been deemed to be lost on the network, wherein the data flow is maintained until data is confirmed to be lost on a the network.

23. (Canceled)

24. (Canceled)

25. (Currently amended) An apparatus according to ~~Claim 23~~ Claim 22, wherein to maintain a data flow on the network in accordance with the adjusted congestion state values is to:

reset, upon receiving an acknowledgement, a limit to the amount of data that a sending host can send before receiving an acknowledgement by adding the size of a data segment that the sending host can send; and

transmitting data on the network.

26. (Currently amended) An apparatus according to ~~Claim 23~~ Claim 22, wherein to re-transmit previously transmitted data when the previously transmitted data has been deemed to be lost on the network is to re-transmit the previously transmitted data upon receiving three duplicate acknowledgements.

27. (Currently amended) An apparatus according to ~~Claim 27~~ Claim 22, wherein the apparatus is to further process a slow-start recovery.

28. (Currently amended) A processor, comprising:

means for adjusting congestion state values, said means for adjusting congestion state values including a means for adjusting a pipe value, wherein the pipe value is based on an estimate of an amount of data outstanding in a network

to a maximum segment size capable of being sent by a sending host plus a difference between a maximum sequence number sent and a lowest sequence number that is not yet acknowledged;

means for maintaining a data flow on a network in accordance with the adjusted congestion state values, wherein data is transmitted if the pipe value is less than a congestion state window; and

means for re-transmitting previously transmitted data when the previously transmitted data has been deemed to be lost on the network.

29. (Original) A processor according to Claim 28, wherein the means for maintaining the data flow on the network in accordance with the adjusted congestion state values re-sets, upon receiving an acknowledgement, a limit to the amount of data that a sending host can send before receiving an acknowledgement by adding the size of a data segment that the sending host can send and continues to transmit data on the network.

30. (Original) A processor according to Claim 28, wherein the means for re-transmitting previously transmitted data re-transmits the previously transmitted data upon receiving three duplicate acknowledgements.